Standalone IgBLAST

Setting up IgBLAST for local analyses of IG sequences

https://ncbi.github.io/igblast/

National Center for Biotechnology Information • National Library of Medicine • National Institutes of Health • Department of Health and Human Services

Introduction

IgBLAST is one of the specialized BLAST services provided by NCBI for use in analyzing and annotate immunoglobulin sequences from model organisms. This service support the analyses of human, mouse, rat, rabbit, and rhesus monkey immunoglobulin sequences and can be accessed at https://www.ncbi.nlm.nih.gov/igblast/. Programs and necessary files for running the searches locally are available as a downloadable package. However, the important germline gene sequence database necessary for the analyses are only available for mouse and rhesus monkey. This handout demonstrates the installation and database configuration through example commands.

Downloading, Installation, and Configuration of IgBLAST

IgBLAST tools have no graphic user interface and only run under command prompt. This demonstration uses a Linux platform, the **wget** and **gunzip** utilities, as well as Perl inline script to simplify/automate file download, archive extraction, and database preparation

It uses the latest version of the IgBLAST standalone package from the NCBI BLAST ftp site:
https://ftp.ncbi.nlm.nih.gov/blast/executables/igblast/release/LATEST/

The following are done in the home directory of a test user account.

Download

```
$ wget -O igblast.tar.gz https://ftp.ncbi.nlm.nih.gov/blast/executables/igblast/release/LATEST/ncbi-igblast
-1.17.1-x64-linux.tar.gz
```

This command uses wget to downloads the package specified by the URL and saves it to a local file with a short name (-O igblast.tar.gz).

Installation

```
$ tar zxpf igblast.tar.gz
```

This command uses tar utility to inflate (z) and then extract (x) all files of the archive (f) while preserving the permission settings (p). It installs the package in the ncbi-igblast-1.17.1 directory with preserved directory structure.

Creating db subdirectory for germline database files

```
$ ls -l ncbi-igblast-1.17.1/
```

The above command examines the setup's directory structure and content.

```
total 60
-rw-r--r-- 1 tao sdesk  4769 Feb 22 17:03 ChangeLog
-rw-r--r-- 1 tao sdesk 27664 Feb 22 17:03 LICENSE
-rw-r--r-- 1 tao sdesk    86 Feb 22 17:03 README
drwxr-xr-x 2 tao sdesk  4096 Feb 22 17:03 bin
drwxr-xr-x 7 tao sdesk  4096 Feb 22 16:15 internal_data
-rw-r--r-- 1 tao sdesk  4679 Feb 22 17:03 ncbi_package_info
drwxr-xr-x 2 tao sdesk  4096 Feb 22 16:15 optional_file

$ mkdir ncbi-igblast-1.17.1/db
$ cd ncbi-igblast-1.17.1/db
```

Germline sequence databases are not distributed with the package. The first command creates a db subdirectory to facilitate the management of database files needed to run the search. The second changes to that subdirectory for database download and preparation.

Downloading Germline Sequence Databases

Getting Preformatted Databases from NCBI

License restriction limits the preformatted germline sequence databases distribution to mouse and rhesus monkey:
https://ftp.ncbi.nlm.nih.gov/blast/executables/igblast/release/database/

```
$ wget https://ftp.ncbi.nlm.nih.gov/blast/executables/igblast/release/database/mouse_gl_VDJ.tar
$ wget https://ftp.ncbi.nlm.nih.gov/blast/executables/igblast/release/database/rhesus_monkey_VJ.tar
```

The above commands download these two databases to the db subdirectory, which is used as the working directory.

Downloading Germline Sequence Databases (cont.)

```
$ tar xpf mouse_gl_VDJ.tar
$ tar xpf rhesus_monkey_VJ.tar
```

The above commands use tar utility to extract the database files from the downloaded archives.

Making Germline Sequence Databases from FASTA
This is a multi-step process, the following is for human sequences from IMGT.

1) Locating and downloading the sequences
The human germline sequences are under the Homo_sapiens subdirectory at IMGT and organized by chain and gene. All files are needed for IgBLAST. To download them in batch, collect file links and save them to a plain text file (ig_imgt_ftp.txt) in the db subdirectory:

```
http://www.imgt.org/download/V-QUEST/IMGT_V-QUEST_reference_directory/Homo_sapiens/IG/IGHD.fasta
http://www.imgt.org/download/V-QUEST/IMGT_V-QUEST_reference_directory/Homo_sapiens/IG/IGHJ.fasta
http://www.imgt.org/download/V-QUEST/IMGT_V-QUEST_reference_directory/Homo_sapiens/IG/IGHV.fasta
http://www.imgt.org/download/V-QUEST/IMGT_V-QUEST_reference_directory/Homo_sapiens/IG/IGKJ.fasta
http://www.imgt.org/download/V-QUEST/IMGT_V-QUEST_reference_directory/Homo_sapiens/IG/IGKV.fasta
http://www.imgt.org/download/V-QUEST/IMGT_V-QUEST_reference_directory/Homo_sapiens/IG/IGLJ.fasta
http://www.imgt.org/download/V-QUEST/IMGT_V-QUEST_reference_directory/Homo_sapiens/IG/IGLV.fasta

$ wget -i ig_imgt_ftp.txt
```

This asks wget to read the URLs in the input file (-i ig_imgt_ftp.txt) and download them sequentially. The command generates console output, which terminated with the following when files are downloaded successfully:

```
FINISHED --2021-07-20 09:14:31--
Total wall clock time: 1.8s
Downloaded: 7 files, 263K in 0.3s (876 KB/s)
```

2) Manipulating the sequence files
The manipulation is to modify the FASTA deflines and strip gaps (represented by dashes) in the sequences. A Perl script (edit_imgt_file.pl) is included in the IgBLAST package to do this. The following Perl one-liner does this for all 7 files in batch:

```
$ perl -e '@f=`ls IG*`; foreach (@f){chomp; $o = "hs_".$_; system("perl ../bin/edit_imgt_file.pl $_ > $o ;
rm $_ ");} exit;'
```

It collects all file names (@f=`ls IG*`;), loops through them (foreach (@f) {...}), and calls the edit_imgt_file.pl to manipulate the file, redirecting output to a renamed file with hs_ prefix and deleting the original file.

3) Making three separate databases for V, J, and D genes
All V sequences need to be combined to make a single V sequence database. The same for J sequences. The following Perl one-liner does this:

```
$ perl -e 'system ("cat hs_*V.fasta > human_gl_V; cat hs_*J.fasta > human_gl_J; cat hs_*D.fasta > hu-
man_gl_D; rm hs_*fasta"); @f=`ls human_gl*`; foreach (@f){chomp; system("../bin/makeblastdb -parse_seqids -
dbtype nucl -in $_ ");} exit;'
```

It combines the relevant genes into a new file and delete the input files afterward (system (“cat …; rm hs_*fasta”);), collects the new output file name (@f=`ls human_gl`;), loops through them (foreach (@f){…}), and format each input as a blast database using the makeblastdb in the bin subdirectory (system(“../bin/makeblastdb -parse_seqids -dbtype nucl -i $_”);). The console output for formatting the germline V gene sequences is:

```
Building a new DB, current time: 07/20/2021 10:21:52
New DB name:   /export/home/tao/ncbi-igblast-1.17.0/db/human_gl_V
New DB title:  human_gl_V
Sequence type: Nucleotide
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 620 sequences in 0.0137119 seconds.
```

Example Use

The setup so far is sufficient to run nucleotide search with query immunoglobulin nucleotide sequences (from mouse, rhesus monkey, and human) against the installed databases for germline gene annotation. The following test runs are for a human example query to demonstrate the key functionality of this package. To avoid mixing working file with installed database files, cd to upper directory first:

```
$ cd ..
```

This moves the working directory from db subdirectory to the parent ncbi-igblast-1.17.1 directory.

Getting a test query sequence

```
$ wget -O Y14934.fna 'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?
db=nuccore&rettype=fasta&id=Y14934'
```

This calls efetch function to retrieve and save the FASTA sequence of Y14934 (a human immunoglobulin record) to a file named Y14934.fna.

Search against the human germline databases

```
$ ./bin/igblastn -query Y14934.fna -organism human -domain_system imgt -germline_db_J ./db/human_gl_J
-germline_db_D ./db/human_gl_D -germline_db_V ./db/human_gl_V -auxiliary_data optional_file/human_gl.aux
-outfmt 3 -show_translation -out Y14934_outfmt3.txt
```

This calls igblastn from the bin subdirectory under current working directory (marked by the dot “.”), uses Y14934.fna as the input query (-query Y14934.fna), with human and imgt annotation scheme (-organism human -domain_system imgt), and search against the specified germline gene databases (all under the ./db subdirectory, thus the path prefix). It asks for “Flat query-anchored, show identities” output format (-outfmt 3) with translation (-show_translation) and saves the results to specified fine (-out Y14934_outfmt3.txt). Use -help option to see all available switches.

```
$ more Y14943_outfmt3.txt
```

This uses more to page through the output file, which lists the annotation summary above the Alignment section:

```
V-(D)-J rearrangement summary for query sequence (Top V gene match, Top D gene match, Top J gene match,
Chain type, stop codon, V-J frame, Productive, Strand, V Frame shift).  Multiple equivalent top matches, if
present, are separated by a comma.
IGHV3-21*01,IGHV3-21*02      IGHD3-22*01    IGHJ4*02      VH     No     In-frame      Yes    +      No

V-(D)-J junction details based on top germline gene matches (V end, V-D junction, D region, D-J junction, J
start).  Note that possible overlapping nucleotides at VDJ junction (i.e, nucleotides that could be
assigned to either rearranging gene) are indicated in parentheses (i.e., (TACT)) but are not included under
the V, D, or J gene itself
AGAGA   TCGGGG  CTATGATAGTAGTGGTTATTAC       GGAAATC TTGAC

Sub-region sequence details (nucleotide sequence, translation, start, end)
CDR3    GTGAGAGATCGGGGCTATGATAGTAGTGGTTATTACGGAAATCTTGACTGC       VRDRGYDSSGYYGNLDC     268     318

Alignment summary between query and top germline V gene hit (from, to, length, matches, mismatches, gaps,
percent identity)
FR1-IMGT       1       54      54      54      0       0       100
CDR1-IMGT      55      78      24      20      4       0       83.3
FR2-IMGT       79      129     51      50      1       0       98
CDR2-IMGT      130     153     24      22      2       0       91.7
FR3-IMGT       154     267     114     111     3       0       97.4
CDR3-IMGT (germline)  268     275     8       7       1       0       87.5
Total   N/A     N/A     275     264     11      0       96
```

The Alignment section shows the alignment with translation and annotation (partial in smaller font to avoid wrapping).

```
                                    <----------------------FR1-IMGT---------------------><-------CDR1-IMGT------><-----------
                                     G  G  G  L  V  K  P  G  G  S  L  R  L  S  C  A  A  S  G  F  P  F  S  N  Y  T  M  H  W  V
                     Query_1     1   GGGGGAGGCCTGGTCAAGCCTGGGGGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATTCCCCTTCAGTAACTACACCATGCACTGGGTC  90
V  96.0% (264/275)  IGHV3-21*01 22  ..........................................................A.........G...T.G....A........  111
                                     G  G  G  L  V  K  P  G  G  S  L  R  L  S  C  A  A  S  G  F  T  F  S  S  Y  S  M  N  W  V
V  96.0% (264/275)  IGHV3-21*02 22  ..........................................................A.........G...T.G....A........  111
V  95.6% (263/275)  IGHV3-21*03 22  ..........................................................A.........G...T.G....A........  111
...                                    -----------------CDR3-IMGT-------------------><-----------FR4-IMGT------------>
```

The complete example result is available on online (as example 1):
https://www.ncbi.nlm.nih.gov/igblast/intro.html

Getting Germline Gene Protein Sequences to Enable igblastp Search

The standalone IgBLAST can be configured to run protein searches.

Downloading sequence files

The protein sequences for germline genes are available from IMGT reference directory, as links in the "F+ORF+in-frame P" column under the "IMGT/GENE-DB reference directory set" section. Since each set of sequences is linked as a query, with the resulted set displayed in a redirected page with generic URL, only manual downloading through copy/paste is possible.

For human, follow the link under Amino acids for each of the three V files (IGHV, IGKV, IGLV), and on the linked page, copy/paste the FASTA sequence into one combined text file. Name the file imgt_hs_gl_v and place it in the /db subdirectory.

Formatting the BLAST database

```
$ cd ~/ncbi-igblast-1.17.1/db
$ perl ../bin/edit_imgt_file.pl imgt_hs_gl_v |  \
  ../bin/makeblastdb -in - -dbtype prot -title  "IMGT human germline V ORF" -parse_seqids \
  -out imgt.human_gl_V
```

The first command changes the working directory to /db subdirectory.
The second is a compound command set. The first part calls the edit_imgt_file.pl (by looking at the parent direction (..) followed by the /bin subdirectory) to work on imgt_hs_gl_v file. The output is passed to makeblastdb (through ("|"). That program will read from the stdin (-in -), makes a protein database (-dbtype prot) with a specified title (-title "string"), parses and indexes the accessions (-parse_seqids, the first string in the defline), and saves the produced db with a specified base name (-out imgt.human_gl_V). The backslash breaks the long command into multiple lines.

The last command generates the following output:

```
Building a new DB, current time: 07/22/2021 21:32:31
New DB name:   /net/lmem21/export/home/tao/ncbi-igblast-1.17.1/db/imgt.human_gl_V
New DB title:  IMGT human germline V ORF
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 624 sequences in 0.279149 seconds.
```

Test search

```
$ cd ..
$ wget -O hs_ig_protein.faa 'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?
db=protein&rettype=fasta&id=AEX29809.1'
$ ./bin/igblastp -query hs_ig_protein.faa -germline_db_V ./db/imgt.human_gl_V \
  -out AEX29809_igblastp_out.txt
```

The commands change the working directory to ncbi-igblast-1.17.1, use efetch to retrieve a sample query sequence and save that to a file (-O hs_ig_protein.faa), and call igblastp to search with this query (-query hs_ig_protein.faa) against the newly created germline gene's ORF database (-germline_db_V ./db/imgt.human_gl_V), and save the result to the specified file (-out AEX29809_ibglastp_out.txt). The Alignment section with domain annotation is shown below.

```
                                    <-------FR1-IMGT-------><CDR1-I><---FR2-IMGT----><CDR2-I><--------------FR3-IMGT----------
                  Query_1       2   VQLVESGGAVVQPGGSLRLSCAASGFNFDDYTMHWVRQAPGRGLEWVSLISWDGGAAYYADSVKGRFTISKDNSKNSLHLQMSSLRTEDT  91
V  90.7% (88/97)  IGHV3-43*01   2   ........V.................T..............K.............ST.............R.......Y...N.......  91
V  88.7% (86/97)  IGHV3-43D*03  2   ........V.................T....A.........K.............ST.............R.......Y...N...A...  91
V  88.7% (86/97)  IGHV3-43D*04  2   ........V.................T....A.........K.............ST.............R.......Y...N...A...  91

                                    ---->
                  Query_1       92  ALYYCAR  98
V  90.7% (88/97)  IGHV3-43*01   92  ......K  98
V  88.7% (86/97)  IGHV3-43D*03  92  ......K  98
V  88.7% (86/97)  IGHV3-43D*04  92  ......K  98
```

Notes

Similar steps can be used to download human T-cell receptor germline sequences, or immunoglobulin sequences for other organisms, and make them into blastable germline databases for use with the standalone igbalst setup.

For T-cell receptor analysis, additional options in iglastn and igblastp needs to be specified explicitly. Use the -help switch to see all available options.

Please send questions and feedback to: blast-help@ncbi.nlm.nih.gov